%{

#include <ctype.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char lexema[60];

void yyerror(char \*msg);

typedef struct {

  char nombre[60];

  double valor;

  int token;

  } tipoTS;

tipoTS TablaSim[100];

int nSim = 0;

int localizaSimb(char \*, int);

void imprimeTablaSim();

int yylex();

int EsPalabraReservada(char[], int);

%}

%token PROGRAMA ID INICIO FIN NUM VARIABLE

%%

S: PROGRAMA ID ';' INICIO listaInstr FIN '.';

listaInstr: instr listaInstr

          |  ;

instr: ID {$$ = localizaSimb(lexema,ID);} ':' '=' expr';';

expr: expr '+' term;

    | expr '-' term;

    | term;

term: term '\*' term2

    | term '/' term2

    | term2;

term2: NUM {$$ = localizaSimb(lexema,NUM);}

     | ID {$$ = localizaSimb(lexema,ID);};

%%

int localizaSimb(char \*nom, int tok) {

  int i;

  for(i = 0; i < nSim; i++) {

    if(!strcasecmp(TablaSim[i].nombre, nom))

      return i;

  }

  strcpy(TablaSim[nSim].nombre, nom);

  TablaSim[nSim].token = tok;

  if(tok == ID) TablaSim[nSim].valor = 0.0;

  if(tok == NUM) sscanf(nom, "%lf", &TablaSim[nSim].valor);

  nSim++;

  return nSim - 1;

}

void imprimeTablaSim(){

  int i;

  for(i = 0; i < nSim; i++) {

    printf("%d  nombre=%s tok=%d valor=%lf\n", i, TablaSim[i].nombre, TablaSim[i].token, TablaSim[i].valor);

  }

}

void yyerror(char \*msg){

  printf("ERROR:%s\n",msg);

}

int EsPalabraReservada(char lexema[], int default\_token) {

  //strcmp considera may y minusc

  //strcasecmp ignora may de min

  if(strcasecmp(lexema,"Program")==0) return PROGRAMA;

  if(strcasecmp(lexema,"Begin")==0) return INICIO;

  if(strcasecmp(lexema,"End")==0) return FIN;

  if(strcasecmp(lexema,"Var")==0) return VARIABLE;

  return default\_token;

}

int yylex(){

  char c;

  int i;

  while(1) {

    c = getchar();

    if(c == ' ') continue;

    if(c == '\t') continue;

    if(c == '\n') continue;

    if(isdigit(c)) {

      i = 0;

      do{

        lexema[i++] = c;

        c = getchar();

      } while(isdigit(c));

      ungetc(c, stdin);

      lexema[i] = '\0';

      return NUM;

    }

    if(isalpha(c)){

      i = 0;

      do{

        lexema[i++] = c;

        c = getchar();

      } while(isalnum(c));

     ungetc(c, stdin);

     lexema[i] = '\0';

     return EsPalabraReservada(lexema, ID);

    }

    return c;

  }

}

int main(){

  if(!yyparse()) printf("La cadena es valida\n");

  else printf("La cadena es invalida\n");

  printf("tabla de simbolos\n");

  imprimeTablaSim();

  return 0;

}



